REMARKS

Claims 21-41 of the application stand rejected. Claim 38 has been canceled herein without prejudice to the filing of continuations and divisionals and Claims 21-37 and 39-41 have been amended to more clearly define the scope of the presently claimed invention. Applicant respectfully requests reconsideration of pending Claims 21-37 and 39-41 in light of the amendments and remarks herein.

Double patenting

The Examiner rejected Claims 21-41 under the judicially created doctrine of obviousness-type double patenting as being unpatentable over Claims 10-18 of U.S. Patent No. 6,138,273. The Examiner suggests that although the conflicting claims are not identical, they are not patentably distinct from each other. Without conceding the appropriateness of the rejection, Applicant is agreeable to filing a terminal disclaimer as suggested by the Examiner in order to address this rejection, upon indication by the Examiner that at least one claim in the present case would be otherwise allowed.

35 U.S.C. §102

Claims 21-22, 26, 28, 32-39 and 41 stand rejected under 35 U.S.C. § 102(b) as anticipated by Peterson (U.S. Patent No. 5,504,901). The Examiner submits that Peterson discloses all the elements of these claims. Applicant respectfully traverses the rejection.

The claims in the present invention are directed at interpreted code and a method and apparatus that provide advantages over typical interpreters. As described in the Specification, Page 6, "The present invention is a virtual processor designed to *remove the transposition* into pseudo code typical of common interpretive engines". This lack of transposition or translation is reflected in each of independent Claims 21, 28, 33 and 36. Thus, the presently claimed invention is distinguishable over both compiled code and typical interpreted code because it eliminates the need for any intermediate code.

Applicant respectfully contends that the scheme in Peterson is directed at procedures (or code) that are compiled and linked, not interpreted, and therefore does not anticipate independent Claims 21, 28, 33 and 36. As described in Peterson:

> "The position independent code instructions for calling another procedure or accessing data that is stored external to the procedure are *generated during compilation and linking* operations and are formulated in a manner that causes the system to determine the location of a called procedure (or the location of data to be accessed) when the instructions are executed (i.e., during run-time execution)....
>
> In the practice of the invention, the instructions for calling another procedure or accessing data that is external to a procedure resolve the memory location of the called procedure (or data to be accessed) based upon the value that is currently stored in the entry point address register and two memory offset pointers that are *established during the compilation and linking of the procedures.*"

Peterson, Col. 4, lines 20-38 (emphasis added)

The sections of Peterson highlighted above support Applicant's contention that the scheme in Peterson includes the steps of compiling and linking. In contrast, of the present invention are directed to creating, *interpreting* and executing a programming language. It is well known to those of ordinary skill in the art that compiled code, as described in Peterson, and interpreted code are significantly different. Specifically, the term "compile" is typically understood to mean transforming a program written in a high-level programming language from source code into object code (intermediate code). The intermediate object code may then be linked to produce machine code (executable code), which may be executed during runtime. "Interpreted" code, on the other hand is generated by an interpreter, i.e., an interpreter typically translates high-level instructions into an intermediate form and then executes the intermediate code. As previously stated, the presently claimed invention is distinguishable over both compiled code and typical interpreted code because it eliminates the need for *any* intermediate code.

Applicants respectfully submit that the sections of Peterson highlighted by the Examiner fail to disclose at least this claimed feature of the invention, namely the feature of encoding instructions without intermediate translations. More specifically, the Examiner points to Peterson, Col. 6 line 60 to Col. 7, line 5 and Col. 7, line 30-50 as teaching the element of encoding the macroinstruction into corresponding subroutine addresses. Applicant respectfully submits that these sections of Peterson do not in fact disclose encoding macroinstructions into corresponding subroutine addresses *without intermediate translations*, as claimed in amended independent Claims 21, 28, 33 and 36.

In fact, the sections of Peterson highlighted by Applicant in the discussion above (Peterson, Col. 4, lines 20-38) show the opposite, namely that Peterson performs various steps during compiling and linking, i.e., *while generating intermediate code*. Thus, if anything, the scheme in Peterson teaches away from the presently claimed invention. Applicant therefore respectfully submits that Peterson does not anticipate independent Claims 21, 28, 33 and 36 and all other claims dependant on these claims, and hereby respectfully requests the Examiner to withdraw the rejection to these claims under 35 U.S.C. §102.

35 U.S.C. §103

Claims 23-25, 29-31 and 40 stand rejected under 35 U.S.C. §103 as being unpatentable over the combination of Peterson in view of Aho ("Compilers, Principles, Techniques and Tools"). Applicant respectfully traverses the Examiner's rejection.

Applicant respectfully submits that Peterson and Aho do not render Claims 23-25, 29-31 and 40 unpatentable. Claims 23-25, 29-31 and 40 are dependant on Claims 21, 28, 33 and 36 and include all features of these independent claims. As previously described, Peterson does not disclose at least one element of the invention as claimed in the independent claims, namely the element of encoding macroinstructions into corresponding subroutine addresses *without intermediate translations*. Combining Aho with Peterson also does not teach or suggest this element. The Examiner highlights Page 65 of Aho as teaching the elements of pushing an argument on a stack, popping an argument from a stack and/or pushing a result onto a stack. Applicant respectfully submits that this is irrelevant to the present discussion. Applicant is not claiming the concepts of pushing an argument on a stack, popping an argument from a stack and/or pushing a result onto a stack in the abstract. Rather, Applicant is claiming these elements as part of a scheme of interpreting code without performing intermediate translations. Applicant thus submits that Peterson and/or Aho, alone or in combination, do not, and in fact cannot, render Claims 23-25, 29-31 and 40 unpatentable and respectfully requests the Examiner to withdraw the rejection to these claims under 35 U.S.C. §103.

CONCLUSION

Based on the foregoing, Applicant respectfully submits that the applicable objections and rejections have been overcome and that pending Claims 21-37 and 39-41 are in condition for allowance. Applicant therefore respectfully requests an early issuance of a Notice of Allowance in this case. If the Examiner has any questions, the Examiner is invited to contact the undersigned at (714) 669-1261.

If there are any additional charges, please charge Deposit Account No. 50-0221.

Respectfully submitted,

Dated: June 21, 2005

Sharmini N. Green
Senior Attorney
Intel Corporation
Registration No. 41,410
(714) 669-1261

Sharmini N. Green
Senior Attorney
Intel Corporation
(714) 669-1261